UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/754,010	01/08/2004	Paul Reuben day	ROC920030366US1	7130

30206 7590 11/17/2009

IBM CORPORATION
ROCHESTER IP LAW DEPT. 917
3605 HIGHWAY 52 NORTH
ROCHESTER, MN 55901-7829

EXAMINER	
LOVEL, KIMBERLY M	

ART UNIT	PAPER NUMBER
2167	

NOTIFICATION DATE	DELIVERY MODE
11/17/2009	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

rociplaw@us.ibm.com

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/754,010
Filing Date: January 08, 2004
Appellant(s): DAY ET AL.

Scott A. Stinebruner (Reg. No. 38,323)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 17 August 2009 appealing from the Office action mailed 17 March 2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Non-Final

The appellant's statement of the status of amendments after non-final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

20040267760	KABRA ET AL	12-2003
20020198867	LOHMAN ET AL	12-2002

Kabra, Navin and David DeWitt. "Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans" ACM (1998), pp 106-117

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 3 and 6 are rejected under 35 U.S.C. 103(a) as being unpatentable over the article "Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans" by Kabra et al (hereafter Kabra) in view of US PGPub 2004/0267760 to Brundage et al (hereafter Brundage).

Referring to claim 1, Kabra discloses a method for automatic handling of errors within a database engine (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*), including the further limitations of:

detecting an error while executing a query access plan [execution plan], and wherein the query access plan is of the type generated by a query optimizer (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan);

in response to detecting the error (see page 109, column 2, line 34 – page 110, column 1, line 4 – after the error is determined the query plan is rebuilt since the remainder of the query plan is based on the estimate), automatically rebuilding the query access plan with query optimizer to generate a new query access plan (see page 110, column 1, lines 2-4 and lines 13-15 – upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan); and

executing the new query access plan to generate at least a portion of a result set for storage or display (see page 110, column 1, line 15 – the fresh new execution plan for the query is executed).

However, Kabra fails to explicitly disclose the further limitation wherein the error is an execution error of a type that halts execution of the query access plan. Brundage

discloses execution of a query (see [0047]), including the further limitations of detecting an error while executing the query, wherein the error is an execution error of a type that halts execution of the query [error is fatal; terminate execution] (see [0183]; [0185]; and [0220]).

It would have been obvious to one of ordinary skill in the art to utilize Kabra's method to re-optimize a sub-optimal query plan to re-optimize the query of Brundage after the fatal error. One would have been motivated to do so in order to improve the performance and efficiency of applications and queries through the generation of optimal plans.

Referring to claim 3, the combination of Kabra and Brundage (hereafter Kabra/Brundage) discloses the method of claim 1, wherein the error is a function check [error in the join] (Kabra: see page 109, column 2, lines 29-33; Brundage: see [0183]; [0185]; and [0220]).

Referring to claim 6, Kabra/Brundage discloses the method according to claim 1, further comprising the step of: reporting the error [message to the user] (Brundage: see [0220]).

Claims 4, 5, 7, 8 and 10-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over the article "Efficient Mid-Query Re-Optimization of Sub-Optimal Query Execution Plans" by Kabra et al in view of US PGPub 2004/0267760 to Brundage et al and further in view of US PGPub 2002/0198867 to Lohman et al (hereafter Lohman).

Referring to claim 4, while Kabra/Brundage discloses the method of claim 1 further comprising the step of: receiving another error while executing a function within the new query access plan (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan), Kabra/Brundage fails to explicitly disclose the further limitations of identifying a first implementation method of the function within the new query access plan; and rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a rebuilt query access plan. Lohman discloses the generation of alternative execution plans (see abstract), including the further limitations of identifying a first implementation method [nested-loop join] of the function within the new query access plan and rebuilding the new query access plan by replacing the first implementation method with a second implementation method [hash join] of the function so as to generate a rebuilt query access plan [LEO's adjustments 130 can cause virtually any physical operator of a QEP to change, and may even alter the structure of the QEP 114] (see [0106] and [0107]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the steps of replacing a function as disclosed by Lohman to modify the execution plan of Kabra/Brundage. One would have been motivated to do so in order to achieve tremendous savings by changing the choice of algorithm being utilized (Kabra: page 109, column 2, Section 2.4 Query Plan Modification, lines 6-11).

Referring to claim 5, Kabra/Brundage fails to explicitly disclose the further limitation of logging information about the error, and the new query access plan.

Lohman discloses the generation of alternative execution plans (see abstract), including the further limitation logging information about the error, and the new query access plan (see [0071]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the logging step disclosed by Lohman in order to record the errors of Kabra/Brundage. One would have been motivated to do so in order to improve the efficiency of the optimizer by utilizing information from past errors.

Referring to claim 7, Kabra discloses a method for automatic handling of errors within a database engine (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*), the method comprising the steps of:

receiving an error while executing a function within a query access plan [execution plan], and wherein the query access plan is of the type generated by a query optimizer (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan);

rebuilding the query access plan with query optimizer to generate a new query access plan (see pages 109, column 2, line 34 – page 110, column 1, line 4 and 110, column 1, lines 2-4 and lines 13-15 – after the error is determined the query plan is rebuilt since the remainder of the query plan is based on the estimate; upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan); and

executing the new query access plan to generate at least a portion of a result set for storage or display (see page 110, column 1, line 15 – the fresh new execution plan

for the query is executed). However, Kabra fails to explicitly disclose the further limitation wherein the error is an execution error of a type that halts execution of the query access plan. Brundage discloses execution of a query (see [0047]), including the further limitations of detecting an error while executing the query, wherein the error is an execution error of a type that halts execution of the query [error is fatal; terminate execution] (see [0183]; [0185]; and [0220]).

It would have been obvious to one of ordinary skill in the art to utilize Kabra's method to re-optimize a sub-optimal query plan to re-optimize the query of Brundage after the fatal error. One would have been motivated to do so in order to improve the performance and efficiency of applications and queries through the generation of optimal plans.

While Kabra/Brundage discloses that the plan may be sub-optimal due to the choice of algorithm being utilized (e.g. hash-join vs. indexed nested-loops join) (Kabra: page 109, column 2, Section 2.4 Query Plan Modification, lines 6-11), Kabra/Brundage fails to explicitly disclose the further limitations of identifying a first implementation method of the function within the query access plan; and rebuilding the query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a new query access plan. Lohman discloses the generation of alternative execution plans (see abstract), including the further limitations of identifying a first implementation method [nested-loop join] of the function within the query access plan and rebuilding the query access plan by replacing the first implementation method with a second implementation method [hash join] of the function

so as to generate a new query access plan [LEO's adjustments 130 can cause virtually any physical operator of a QEP to change, and may even alter the structure of the QEP 114] (see [0106] and [0107]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the steps of replacing a function as disclosed by Lohman to modify the execution plan of Kabra/Brundage. One would have been motivated to do so in order to achieve tremendous savings by changing the choice of algorithm being utilized (Kabra: page 109, column 2, Section 2.4 Query Plan Modification, lines 6-11).

Referring to claim 8, Kabra/Brundage/Lohman discloses the method of claim 7, wherein the function is one of a join function [error in the join], an indexing function, a grouping function, and an ordering function (Kabra: see page 109, Section 2.4 Query Plan Modification, lines 7-10; Brundage: see [0107]).

Referring to claim 10, while Kabra/Brundage receiving another error while executing the function within the new query access plan (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan), Kabra/Brundage fails to explicitly disclose the further limitation of rebuilding the new query access plan by replacing the second implementation method with a third implementation method of the function. Lohman discloses the generation of alternative execution plans (see abstract), including the further limitations of identifying a second implementation method [nested-loop join] of the function within the new query access plan and rebuilding the new query access plan by replacing the second implementation method with a third implementation method [hash join] of the function

[LEO's adjustments 130 can cause virtually any physical operator of a QEP to change, and may even alter the structure of the QEP 114] (see [0106] and [0107]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the steps of replacing a function as disclosed by Lohman to modify the execution plan of Kabra/Brundage. One would have been motivated to do so in order to achieve tremendous savings by changing the choice of algorithm being utilized (Kabra: page 109, column 2, Section 2.4 Query Plan Modification, lines 6-11).

While the combination of Kabra/Brundage and Lohman (hereafter Kabra/Brundage/Lohman) fails to explicitly disclose that another error occurs, it would be obvious to one of ordinary skill in the art to apply the same steps to utilized to re-optimize the first sub-optimal plan to optimize a second sub-optimal plan since this is merely a repetitive step that occurs repetitively with any query plan.

Referring to claim 11, Kabra/Brundage/Lohman discloses the method according to claim 10 further comprising the step of: logging information about the error, the another error, and the new query access plan (Lohman: see [0071]).

Referring to claim 12, Kabra discloses a method for automatic handling of errors within a database engine (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*), the method comprising the steps of:

executing a query access plan comprising a plurality of functions, each function including a first implementation method, and the query access plan of the type generated by a query optimizer (see page 109, column 2, lines 34-37; page 110, column 1, 10-15; and Fig 4);

detecting a first error while executing a first function within a query access plan [execution plan] (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan);

rebuilding the query access plan with query optimizer to generate a new query access plan (see pages 109, column 2, line 34 – page 110, column 1, line 4 and 110, column 1, lines 2-4 and lines 13-15 – after the error is determined the query plan is rebuilt since the remainder of the query plan is based on the estimate; upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan); and

executing the new query access plan to generate at least a portion of a result set for storage or display (see page 110, column 1, line 15 – the fresh new execution plan for the query is executed). However, Kabra fails to explicitly disclose the further limitation wherein the error is an execution error of a type that halts execution of the query access plan. Brundage discloses execution of a query (see [0047]), including the further limitations of detecting an error while executing the query, wherein the error is an execution error of a type that halts execution of the query [error is fatal; terminate execution] (see [0183]; [0185]; and [0220]).

It would have been obvious to one of ordinary skill in the art to utilize Kabra's method to re-optimize a sub-optimal query plan to re-optimize the query of Brundage after the fatal error. One would have been motivated to do so in order to improve the performance and efficiency of applications and queries through the generation of optimal plans.

While Kabra/Brundage discloses that the plan may be sub-optimal due to the choice of algorithm being utilized (e.g. hash-join vs. indexed nested-loops join) (Kabra: page 109, column 2, Section 2.4 Query Plan Modification, lines 6-11) and receiving an error during execution, Kabra/Brundage fails to explicitly disclose the further limitation of rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function. Lohman discloses the generation of alternative execution plans (see abstract), including the further limitation of rebuilding the query access plan by replacing the first implementation with a second implementation method [hash join] of the function [LEO's adjustments 130 can cause virtually any physical operator of a QEP to change, and may even alter the structure of the QEP 114] (see [0106] and [0107]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to utilize the steps of replacing a function as disclosed by Lohman to modify the execution plan of Kabra/Brundage. One would have been motivated to do so in order to achieve tremendous savings by changing the choice of algorithm being utilized (Kabra: page 109, column 2, Section 2.4 Query Plan Modification, lines 6-11).

While Kabra/Brundage/Lohman fails to explicitly disclose that a second error occurs, it would be obvious to one of ordinary skill in the art to apply the same steps to utilized to re-optimize the first sub-optimal plan to optimize a second sub-optimal plan since this is merely a repetitive step that occurs repetitively with any query plan.

(10) Response to Argument

This Examiner's Answer will address the Appellant's Arguments in the order in which they appear in the appeal brief.

- ***Issue A: Claims 1, 3 and 6 are not patentable over Kabra in view of Brundage***

  - ***Independent claim 1***

Appellant's Argument: Claim 1 recites a method for automatic handling of errors within a database engine, which includes detecting an error while executing a query access plan; in response to detecting the error, automatically rebuilding the query access plan with the query optimizer to generate a new query access plan; and executing the new query access plan to generate at least a portion of a result set for storage or display. The claim also recites that the error is an execution error of a type that halts execution of the query access plan, and that the query access plan is of the type generated by a query optimizer.

Notably, the error that is addressed in claim 1 is specifically an execution error of a type that halts execution of the query access plan. …

In rejecting claim 1, the Examiner relies on Kabra and Brundage. The Examiner asserts that Kabra discloses limitations of the claim 1 at the abstract, lines 6-8 and page

109, col. 2, line 34 to page 110, col. 1, line 15. The Examiner admits, however, that Kabra fails to explicitly disclose the limitation "wherein the error is an execution error of a type that halts execution of the query access plan," required by claim 1. For this, the Examiner now cites Brundage, and specifically paragraphs [0047], [0183], [0185] and [0220] thereof. … And although the Examiner's new reference Brundage does in fact disclose the concept of a fatal error (see, e.g., "the error is fatal (re: terminate execution)" at paragraph [0220] of Brundage), Applicant respectfully submits that the combination of Kabra and Brundage still does not disclose or suggest all of the limitations in the same manner claimed. First, the mere mention in Brundage of the concept of a fatal error adds little to the rejection. Applicant has already acknowledged that execution errors of the type that halt execution of a query are known in the Background portion of the specification (see, e.g., page 3, lines 17-21 of the Application). … Brundage's disclosure of these types of errors therefore adds little to the known state of the art as it pertains to claim 1. … However, these engines do not have any ability to repair or correct the error without manual intervention. As a result, in response to such an error, the next step for the database user typically involves calling a customer support engineer and trying to resolve the problem via telephone or e-mail, leading to user frustration, system downtime, and lower productivity. … Brundage does not disclose or suggest any functionality capable of catching errors that halt execution and then handling such errors by recompiling and continuing with query execution. In addition, with respect to Kabra, Applicant submits that execution errors are entirely different in kind from the types of "errors" contemplated in Kabra (sub-optimally

executing queries), so it is not a mere obvious extension of Kabra to apply its disclosed method to address execution errors. Therefore, combining Brundage with Kabra, which merely discloses detection of a sub-optimal query and voluntary modification or re-optimization of the same (and thus operating only on events that are not analogous to execution-type (fatal) errors), the proposed combination does not disclose or suggest any type of error handling facility for execution-type (fatal) errors that avoids having to halt execution of a query by automatically rebuilding and executing the access plan for the query. Applicant therefore respectfully submits that the combination of Kabra and Brundage still does not disclose each and every limitation of claim 1. (Pages 8-10 of the Remarks)

Examiner's Response: The examiner respectfully disagrees that the proposed combination of Kabra and Brundage fails to disclose the claimed invention. As pointed out in the rejection of claim 1, Kabra discloses a method for automatic handling of errors within a database engine (see abstract, lines 6-8 – the sub-optimality is considered to represent the *error*), including the further limitations of:

detecting an error while executing a query access plan [execution plan], and wherein the query access plan is of the type generated by a query optimizer (see page 109, column 2, lines 34-37 and page 110, column 1, 10-15 – the error is found during execution of the execution plan);

in response to detecting the error (see page 109, column 2, line 34 – page 110, column 1, line 4 – after the error is determined the query plan is rebuilt since the remainder of the query plan is based on the estimate), automatically rebuilding the

query access plan with query optimizer to generate a new query access plan (see page 110, column 1, lines 2-4 and lines 13-15 – upon the determination that the plan is sub-optimal, the query optimizer is re-invoked to generate a new execution plan); and

executing the new query access plan to generate at least a portion of a result set for storage or display (see page 110, column 1, line 15 – the fresh new execution plan for the query is executed).

Therefore, Kabra is considered to teach the concept of automatically handling an error during execution of the query access plan and then automatically rebuilding the query access plan with the query optimizer to generate a new plan. As pointed out in the rejection, the query optimizer is re-invoked without user intervention. The examiner construes the sub-optimality of the plan to represent an error. The examiner agrees with the Appellant that Kabra fails to disclose the concept of the error halting execution of the query. However, Brundage in paragraph [0220] discloses the concept of an error which halts execution of a query. It would have been obvious to one of ordinary skill in the art to try the system of Kabra which automatically rebuilds a query plan with an error, to rebuild a query plan with the type of error disclosed by Brundage.

Appellant's Argument: Second, it is only through the benefit of hindsight and through the prism of Applicant' s disclosure that the Examiner can extrapolate that the combination of Kabra and Brundage discloses or suggests claim 1. In particular, Applicant respectfully disagrees with the Examiner' s argument at page 5 that Kabra and Brundage are combinable, namely, "[i]t would have been obvious to one of ordinary

skill in the art to utilize Kabra' s method to re-optimize a sub-optimal query plan to re-optimize the query of Brundage after the fatal error...to improve the performance and efficiency of applications through the generation of optimal plans." There is nothing in either reference that would suggest to one of ordinary skill in the art that Kabra' s re-optimization method could be used to address execution-type errors. Kabra, as admitted by the Examiner, does not suggest any applicability of its method to anything other than sub-optimally of executing queries. Furthermore, the mere mention of fatal errors in Brundage, without any disclosure whatsoever regarding how such errors are handled, does little or nothing to address this shortcoming in Kabra. In fact, Applicant can find no particular relevance of Brundage to the claim beyond a simple keyword match with the term "fatal error" since the reference is otherwise unconcerned with error handling. There must be some reasonable expectation that one of ordinary skill in the art would be motivated to use Kabra's optimization method to address instances where a query execution must be halted as the result of encountering an execution error, and neither Kabra nor Brundage provides any objective evidence of any such motivation. In fact, neither reference even addresses how execution errors of the type recited in claim 1 should be handled. As noted above, the differences between a sub-optimal query (which does not require the query to be halted) and an execution error (which conventionally requires execution to be halted) are such that it is not a mere obvious modification to retrofit the Kabra process to address execution errors in the same way as sub-optimal queries are addressed. The Kabra process, for example, optimizes sub-optimal queries by reallocating system resources or revising a query access plan for

improved performance (e.g., by revising a join order or table access method). There has been no evidence proffered that there would be any reasonable expectation by one of ordinary skill in the art that such operations would address execution errors that would otherwise completely halt execution of a query. Therefore, given the fundamental principles behind the respective implementations of the references and Applicant's claim 1 are so different, it would only be through the benefit of Applicant's disclosure that one of ordinary skill in the art would ever contemplate taking the disparate pieces from the Kabra and Brundage designs and combining them together in the manner suggested by the Examiner. Such is the essence of hindsight-based analysis, and is antithetical to the requirements for establishing a primafacie case of obviousness. (Pages 10-11 of the Remarks)

Examiner's Response: In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). It would have been obvious to one of ordinary skill in the art to utilize the program code which detects and rebuilds plans that are sub-optimal with plans that contain other types of errors. One type of error is merely being replaced by a different

type of error, however the outcome of a rebuilt plan is the same. Furthermore, both Kabra and Brundage are in the field of endeavor of query execution.

- ***Dependent Claim 3***

Appellant's Argument: Claim 3 depends from claim 1 and additionally recites that the error is a function check. The Examiner cites Kabra at page 109, column 2, lines 29-33 and Brundage at paragraphs [0183], [0185], and [0220] for allegedly disclosing this feature. However, the cited disclosure of Kabra at page 109 does not disclose a function check, which is recognized in the art, and specifically defined in the specification, as a type of error that halts execution in a database engine. Moreover, the Examiner's assertion that Kabra discloses a function check is completely inconsistent with the Examiner' s admission (made in connection with the rejections of claims 1, 7 and 12) that Kabra does not disclose an error that halts execution of a query plan. In addition, while Brundage discloses "fatal errors," Brundage does not specifically disclose the concept of a "function check," which by virtue of claim differentiation, is required to be a specific type of "execution error of a type that halts execution of the query access plan." (page 12 of the Remarks)

Examiner's Response: The examiner respectfully disagrees that the combination of Kabra and Brundage fails to teach the concept of a function check. As pointed out on page 3 under the heading "Claim Objections" of the Non-Final Rejection mailed 17 March 2009, the specification fails to explicitly define the term "function

check" and therefore the intentions of the limitation are unclear. It is unclear whether the limitation relates to a function that checks for errors or an error checking mechanism that is actually checking for an error within the function. Page 10, lines 3-6 of the Applicant's Specification states the following "During execution, the database engine 44 may in step 304, detect an error that halts execution of the query. Within some environment, such an error is known as a function check. A number of different errors may be detected …." Furthermore, page 11, lines 17-19 of Applicant's Specification states "When a query is running, in step 310, and an error is detected, in step 316, that error may indicate that there is a problem with performing a particular function of the query, such as, for example, the grouping function." Therefore, since the Specification fails to explicitly define the term "function check," the examiner construes the phrase as meaning checking for an errors within a function. Page 109, column 2, lines 27-33 of Kabra states "While that can result in significant savings in some cases, a much more serious problem with query execution is that the query execution plan itself may be sub-optimal. For example, the join order might be sub-optimal, or the choice of algorithms (e.g., hash-join vs. indexed nested loops join could be improved." In Kabra, the examiner construes the sub-optimality of query access plan caused by the join order or choice of algorithms as representing the claimed error. The join order and algorithms are considered to represent the functions. Therefore, given the broadest reasonable interpretation of the claimed limitation, the sub-optimality of the query due to the join order or algorithms is considered to teach the concept of a function check. While the error of Kabra fails to halt execution of the query, Brundage as stated in the rejection of

claim 1, teaches the concept of an error which halts execution of a query in paragraph [0220].

- ***Dependent Claim 6***

Appellant's Argument: Claim 6 depends from claim 1 and additionally recites reporting the error. The Examiner cites Brundage at paragraph [0220] for allegedly disclosing this feature. However, as argued above in connection with claim 1, Applicant respectfully submits that when claim 6 is read in context with the limitations of claim 1 from which it depends, the Examiner' s rejection based on the combination of Kabra and Brundage cannot be sustained despite the disclosure in Brundage that "the first must be a string sequence, and describes a message to the user." Reversal of the Examiner's rejection of claim 6, and allowance of the claim, are therefore respectfully requested for these reasons and the reasons set out in connection with claim 1.

Examiner's Response: The rejections of claim 6 is maintained for the reasons stated above in regards to claim 1.

- ***Issue B: Claims 4, 5, 7, 8 and 10-12 are not patentable over Kabra in view of Brundage and further in view of Lohman***

- ***Independent Claim 7***

Appellant's Argument: Claim 7 generally recites a method for automatic handling of errors within a database engine. This method includes receiving an error while executing a function within a query access plan. The error is an execution error of a type that halts execution of the query access plan, and the query access plan is of the type generated by a query optimizer. The method also includes identifying a first implementation method of the function within the query access plan, rebuilding the query access plan with the query optimizer by replacing the first implementation method with a second implementation method of the function so as to generate a new query access plan, and executing the new query access plan to generate at least a portion of a result set for storage or display. In rejecting claim 7, the Examiner also relies on the same arguments and citations of Kabra and Brundage made in connection with claim 1. The Examiner admits, however, that the combination of Kabra and Brundage does not disclose "identifying a first implementation method of the function within the new query access plan" and "rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a rebuilt query access plan". For the latter two limitations, the Examiner cites Lohman (U.S. Patent Application Publication No. 2002/0198867), and in particular, paragraphs [0106] and [0107]. Lohman (U.S. Patent Application Publication No. 2002/0198867), like Kabra, focuses on improving query optimization, in other words, improving the execution of sub-optimal queries. Lohman (U.S. Patent Application Publication No. 2002/0198867) accomplishes this by using empirical measurements from the query plan chosen for execution to validate if the model or estimates was

correct or erroneous (e.g., claim 1 of Lohman (U.S. Patent Application Publication No. 2002/0198867)). If the model is in error, then one or more adjustments to the model are made to correct the error. However, like Kabra, the error in Lohman (U.S. Patent Application Publication No. 2002/0198867) has nothing to do with an error that halts execution; instead this error leads to sub-optimal plan selection. And although paragraphs [0106] and [0107] mention replacing a nested-loop join with a hash join, this is in the context of improving sub-optimal access plans, akin to Kabra, and does not disclose the remaining limitations in the same manner claimed. (Pages 13-14 of the Remarks)

Examiner's Response: As stated above in response to the arguments in regards to claim 1, the combination of Kabra and Brundage are considered to teach the concept of an error which halts execution of the query. As pointed out in the rejection of claim 7 and by the Appellant, Lohman discloses the generation of alternative execution plans (see abstract), including the further limitation of rebuilding the query access plan by replacing the first implementation with a second implementation method [hash join] of the function [LEO's adjustments 130 can cause virtually any physical operator of a QEP to change, and may even alter the structure of the QEP 114] (see [0106] and [0107]). Therefore, the combination of Kabra, Brundage and Lohman is considered to meet the requirements of the claim language.

- ***Independent Claim 12***

Appellant's Argument: In particular, none of Kabra, Brundage and Lohman discloses or suggests rebuilding a query access plan with a query optimizer to generate a new query access plan and executing the new query access plan to generate at least a portion of a result set for storage or display, in response to an execution error of the type that halts execution of a query. Moreover, none of the references addresses any functionality for handling a subsequent error that occurs after a query access plan has been rebuilt and re-executed. The Examiner merely discounts this additional functionality, but the fact remains that none of the references discloses or suggests a multi-level response strategy that attempts to address execution errors in the hierarchical fashion recited in claim 12. As is even acknowledged by Kabra in connection with handling sub-optimal queries, re-optimizing queries comes with a performance cost, so the costs of any error handling processes must be weighed against potential benefits. Claim 12 is directed to a process whereby different operations are performed the second time an error is received than a first time, with the expectation that errors can be handled in an efficient and competent manner. None of the art cited by the Examiner appreciates the desirability of such a multi-step approach. (Pages 14-15 of Remarks)

Examiner's Response: The rejections of claim 12 are maintained for the reasons stated above in regards to claims 1 and 7. Furthermore, while

Kabra/Brundage/Lohman fails to explicitly disclose that a second error occurs, it would be obvious to one of ordinary skill in the art to apply the same steps to utilized to re-optimize the first sub-optimal plan to optimize a second sub-optimal plan since this is merely a repetitive step that occurs repetitively with any query plan. Performing the step a second time is considered to represent mere duplication of steps.

- ***Dependent Claims 4 and 10***

Appellant's Argument: Claims 4 and 10, which depend from claims 1 and 7, respectively, recite to varying extents the concept of handling another error detected while executing a query access plan that has been automatically rebuilt in response to an execution error by rebuilding the query access plan to replace a first implementation method of a function with a second implementation method. Claim 4 is representative, and recites receiving another error while executing a function within the new query access plan, identifying a first implementation method of the function within the new query access plan, and rebuilding the new query access plan by replacing the first implementation method with a second implementation method of the function so as to generate a rebuilt query access plan. Claim 10 adds the concept of replacing a second implementation of a function with a third implementation in response to an additional error. In rejecting claim 4, the Examiner relies on paragraphs [0106] and [0107] of Lohman (U.S. Patent Application Publication No. 2002/0198867). In addition, as noted above in connection with claim 12, irrespective of whether any of the operations

disclosed in Kabra, Brundage, and Lohman (U.S. Patent Application Publication No. 2002/0198867) can be analogized to functions or function implementations, none of the references discloses a multi- stage error handling protocol capable of handling multiple errors that are encountered during the execution of a query plan. (Pages 15-16 of the Remarks)

Examiner's Response: The rejections of claims 4 and 10 are maintained for the reasons stated above in regards to claim 12.

- ***Dependent Claims 5 and 11***

Appellant's Argument: Claims 5 and claim 11, which depend from claims 1 and 7, respectively, recite to varying extents logging information about at least one error and the new query access plan. Claim 11 is representative, and recites logging information about the error, the another error, and the new query access plan. In rejecting claim 11, Examiner relies on paragraph [0071] of Lohman (U.S. Patent Application Publication No. 2002/0198867). However, paragraph [0071] of Lohman refers to an old estimate, a new estimate, etc. The passage does not disclose logging information about errors and the new query access plan. Logging estimates or statistics is NOT the same as logging information about errors and the query access plan. (Page 16 of the Remarks)

Examiner's Response: The examiner respectfully disagrees that Lohman fails to disclose the concept of logging information about the errors of the plan. In Lohman, the errors are considered to be the sub-optimality of the plan. The statistics are

considered to represent information in regards to the sub-optimality of the plan. Therefore, logging the estimates is considered to represent logging information about the error.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Kimberly Lovel

/Kimberly Lovel/

Examiner, Art Unit 2167

Conferees:

/John R. Cottingham/

Supervisory Patent Examiner, Art Unit 2167

/James Trujillo/

Supervisory Patent Examiner, Art Unit 2159